Distributed Systems

* High Performance Distributed Computing
  + Cluster Computing
    - Computer comprised of other multiple computing nodes with each node having the same operating system.
    - Example: The CofC compute cluster
  + Grid Computing
    - A computing cluster where each node operates on a separate domain.
    - Example: a radio telescope array
    - Layers of architecture
      * Fabric layer
      * Connectivity and Resource layer
      * Collective layer
      * Application layer (last layer)
  + Cloud Computing
    - Example: AWS Marketplace
    - Separated and organized by four layers
      * Hardware – physical routers, servers, storage, etc.
      * Infrastructure – VMs
      * Platform – operating systems
      * Application – cloud user-end implementation
* Distributed Information Systems
  + Distributed Transaction Processing
    - Remote programs are referred to as clients.
    - Sub transactions joined together form a nested transaction.
    - Transactions: BEGIN\_TRANSACTION and END\_TRANSACTION
      * No partial commits only rollbacks
      * ACID
        + Atomic – No partial commits. Either the total instruction executes, or none of them do.
        + Consistent – the database should be consistent before and after the transaction.
        + Isolated – There needs to be logical isolation. The transactions should execute alone or with others the same way.
        + Durable – Whatever changes made within the system need to last
  + Enterprise Application Integration (EAI)
    - Remote Procedure Calls (RPC) and Remote Method Invocations (RMI)
      * RPC operates on functions, whereas RMI works on objects.
    - Message Oriented Middleware (MOM)
      * MOM translates an output from a specific location to an input in another location
    - Four ways to integrate applications
      * File transfer
      * Shared database
      * Remote procedure call
      * Messaging
  + Example: Facebook Adsense
* Pervasive Systems – naturally distributed
  + Sensors and actuators
    - Sensors read and analyze the user’s behavioral patterns.
    - Actuators are what give the necessary information and feedback needed in order for the sensors to be able to work effectively and efficiently.
    - Example: Fitbit counting your steps and alerting yo
  + Ubiquitous Computing Systems
    - Main Requirements
      * Distribution:
        + Distribution should come naturally and transparent to the user.
      * Interaction:
        + A user is not fully aware of the input to a specific computer system they are using
      * Context Awareness:
        + Characterized by the who, what, where, when, and what. The proper sensors must be available in order to detect the contextual information
      * Autonomy:
        + Responsive to changes in the system without the need for human interaction
      * Intelligence:
        + Able to make new decisions off of new information
    - Example: Uber’s use of smart phones for ride hailing and location services
  + Mobile Computing Systems
    - Must be able to function on a wireless network.
    - Disruption-tolerant networks
      * continual data integrity between separate node connectivity
    - Pocket-Switched Networks
      * They are not always connected, and a packet can reach a destination using either user mobility or network connectivity - like transitioning from 4G to WiFi
    - Example: Smartphones
  + Sensor Networks
    - Networks with many small nodes containing at least one sensor.
    - Abstract region makes a node able to recognize a certain address from a group of nodes from where it can collect information.
    - In-network data processing allows for complex processing of data within a network.
    - Example: Drone Swarms